Comparaison des cibles d'une matrice de score et d'une expression régulière. Application à la recherche de sites de fixation du facteur de transcription LXR

Aymeric Antoine-Lorquin, Sandrine Lagarrigue, Frédéric Lecerf, Jacques Nicolas, Catherine Belleannée

Comparison of the targets obtained by a scoring matrix and by a regular expression. Application to the search for LXR binding sites.

Aymeric ANTOINE-LORQUIN[1], Sandrine LAGARRIGUE[2,3], Frédéric LECERF[2,3], Jacques NICOLAS[1] and Catherine BELLEANNÉE[1]

[1] Irisa/Inria/Université de Rennes1, Campus de Beaulieu, 35042 Rennes, FRANCE
[2] INRA, UMR 598 Génétique Animale, F-35000 Rennes, France
[3] Agrocampus Ouest, UMR 598 Génétique Animale, F-35000 Rennes, France

Corresponding Author: aymeric.antoine-lorquin@irisa.fr

Abstract *In bioinformatics, it is a common task to search for new instances of a pattern built from a set of reference sequences. For the simplest and most frequent cases, patterns are represented in two ways : regular expression or scoring matrix. In the first case, the acceptance of a sequence is a binary decision. In the second case, the quality of the sequence is indicated by a score. Since both representations seem to be used indifferently in pratice, one may wonder if they have any impact on the result. Is there a best representation? What is the accurate threshold value for a scoring matrix? Allowing mutations in a regular expression is it comparable to moving the score of acceptance of a matrix? These are questions adressed in this paper, through a test case on binding site search. This study compares hits obtained with scoring matrices or by regular expressions allowing up to two substitutions. The study shows that, in our LXR study, sequences found by a scoring matrix are closer to the targeted hits than sequences found by a regular expression.*

Keywords pattern matching, position-weight matrix, approximate regular expression, pattern, RSAT, transcription factor binding sites

Comparaison des cibles d'une matrice de score et d'une expression régulière. Application à la recherche de sites de fixation du facteur de transcription LXR

Résumé *En bio-informatique, il est habituel de rechercher de nouvelles instances d'un modèle construit à partir d'un ensemble de séquences de référence. Dans la majorité des cas, les plus simples, ces modèles sont représentés soit par des expressions régulières, soit par les matrices de score. Dans le cas des expressions régulières, le résultat d'une analyse est binaire (acceptation ou rejet). Dans le cas des matrices de score, un score indique la qualité du résultat. Si, en pratique, ces deux représentations semblent pouvoir être utilisées indifféremment, on peut se demander si elles ont un impact sur le résultat. Y'a-t-il une meilleure représentation ? Comment fixer le seuil d'acceptabilité d'une matrice de score ? Autoriser des mutations sur une expression régulière est-il comparable à faire varier le seuil d'acceptation d'une matrice? Ce sont des questions évoquées dans ce papier, au travers du cas d'application du site de fixation de LXR. Cette étude compare les occurrences obtenues avec une matrice de score et avec une expression régulière autorisant jusqu'à deux substitutions. Elle montre que, dans notre étude LXR, les séquences obtenues avec une matrice de score sont plus proches des références que les séquences obtenues par l'expression régulière.*

Mots-clés *pattern matching*, matrice poids-position, expression régulière approchée, motif, RSAT, site de fixation de facteur de transcription

1. Introduction

Une problématique classique en bio-informatique consiste en la recherche d'éléments proches de séquences de référence. Par exemple, dans le cas des séquences cibles des facteurs de transcription (Transcription Factor Binding Sites, TFBS), on peut disposer d'une série d'occurrences connues et vouloir rechercher leurs répliques exactes, mais aussi des séquences légèrement mutées, les TFBS ayant des motifs labiles.
Dans le cas de motifs simples comme ceux des TFBS, il existe essentiellement deux familles d'approches pour représenter et chercher un motif à partir de références connues. La première est celle des *expressions régulières* :

elle consiste à indiquer quelles sont les bases permises à chaque position de la séquence, par le biais de formats plus ou moins élaborés (ex : dreg (EMBOSS) [1], grappe [2], scripts NRgrep [3]). Les expressions régulières aboutissent à une réponse binaire : acceptation ou rejet d'une occurrence. En fait, ce type de représentation revient à indiquer sous une forme compacte l'énumération des variants acceptés.
La seconde méthode est celle de type *matrice de score* (ex : TESS [4], RSAT [5], oPOSSUM [6]), qui consiste globalement à affecter un poids à chaque lettre pour chaque position en fonction de sa représentativité dans les séquences de référence. Contrairement aux expressions régulières, il ne s'agit pas ici de valider directement la présence d'une occurrence dans une séquence mais d'attribuer un score de similarité à chaque occurrence potentielle. Le choix est ensuite laissé à l'utilisateur de décider du seuil d'acceptation des occurrences.

Ces deux types d'outils semblant utilisés indifféremment dans la pratique, on s'est demandé si choisir l'un ou l'autre avait ou non un impact sur les résutats. Le côté binaire de la décision par l'expression régulière peut être jugé un peu radical et amener à privilégier les signatures à base de score, telles que les matrices poids-position. Qu'en est-il réellement ? En pratique, quelles sont les occurrences retenues par la matrice ? Comment fixer un score utilisateur pertinent pour valider ou rejeter une occurrence ? De bons candidats (i.e. proches des références) peuvent-ils passer à travers les mailles de l'une ou l'autre méthode ? Le fait d'autoriser des substitutions sur les expressions régulières (*approximate regular expression matching* [7]) permet-il le même genre de graduation dans les résultats trouvés que le fait de faire varier un score seuil sur une matrice ?

L'approche suivie pour cette étude est avant tout pratique. Ainsi, nous explorons les différences entre les deux types de représentation, expressions régulières et matrices de score, au travers de l'analyse d'un cas particulier pour lequel nous disposons d'une expertise biologique : la recherche de sites de fixation du facteur de transcription LXR[8]. Ce cas correspond à un cas réel, possédant une divergence "raisonnable" entre les séquences de référence. Le logiciel de *pattern matching* utilisé pour la matrice de score est RSAT[5], qui est un outil mature, très utilisé et bien maintenu.

Nous avons analysé quantitativement et qualitativement les occurrences acceptées par chacune des méthodes. Pour cela deux séries de jeux de données ont été constituées. Toutes les données sont des séquences de taille 16 (i.e. la longueur du motif). La série ERx correspond aux séquences acceptées par l'expression régulière à deux substitutions près. On va chercher à savoir dans quelle mesure elles correspondent également aux séquences ciblées par la matrice. La deuxième série, Refx, correspond aux séquences proches des séquences de référence (les sites avérés). Elle représente quant-à-elle les séquences qu'on cherche à cibler.

2. Construction d'un motif LXR : matrice de score versus expression régulière

2.1 Sites LXR : Sur le cas d'application LXR, nous disposons de 13 séquences de référence, validées expérimentalement comme site de fixation du facteur LXR (cf TABLE 1). La finalité biologique est de découvrir de nouveaux sites putatifs du LXR dans les génomes de différentes espèces en s'inspirant de ces sites connus. D'après l'expertise biologique, le site LXR est de type DR4, c'est-à-dire que les 4 nucléotides centraux influencent peu la fixation du facteur. C'est pourquoi leurs scores sont considérés équilibrés (cf Table2 et Fig1).

2.2 Matrice mLXR : Dans le cas d'une matrice poids-position, les nucléotides de chaque référence pour chaque position sont comptabilisés pour calculer le poids d'une lettre en fonction de sa position. Plus une lettre est représentée, plus elle a un poids important et c'est la somme des poids d'une occurrence qui détermine son score final. Par exemple, dans la matrice mLXR (cf TABLE 2), à la position 1, T est le nucléotide majoritaire avec un poids de 10, tandis que C est minoritaire avec un poids de 3 : une occurrence commençant par T aura donc un meilleur score final qu'une même occurrence commençant par C. Par ailleurs, à cette même position, A et G ont un poids de 0 : ce sont donc apparemment des lettres "interdites" pour cette position. Or, une occurrence commençant par l'un ou l'autre de ces nucléotides aura un score final plus faible qu'une même occurrence commençant par C ou T, mais il pourra quand même être retenu si le score total de l'occurrence est supérieur au seuil fixé. Sur cette matrice, on considère une équirépartition des lettres pour les quatre positions centrales (7 à 10) : ainsi, la nature de la lettre à cette position de l'occurrence n'aura pas d'incidence sur le score final. Les versions élaborées de calcul de score peuvent de plus prendre en compte le bruit de fond nucléotidique afin de pondérer les scores en fonction de la représentativité individuelle des nucléotides dans la séquence analysée (ex : RSAT [5]).

Id	Ensembl Id	Species	Sequence		Score RSAT
01	Cyp7Alpha1	Souris	TGAACTtgggTGACCA		9,7
02	Cyp7Alpha1	Rat	TGAACTtgagTGACCA		9,7
03	FASN	Souris	TGACCGgtagTAACCC		13,7
04	FASN	Rat	TGACCGgtagTAACCC		13,7
05	FASN	Poule	TGACCTgtggTAACCT		12,9
06	FASN	Humain	TGACCGgcagTAACCC		13,7
07	LPCAT3	Humain	CGACCGggagTAACCT		12,4
08	LPCAT3	Souris	CGACCGagagTAACCT		12,4
09	LPCAT3	Rat	CGACCGagagTAACCT		12,4
10	LPCAT3	Poule	TGCCCGcgagTAACCC		12,1
11	CETP	Humain	TGCCCGacaaTGACCC		11,3
12	CYP51a	Humain	TGACCTcaggTGATCC		10
13	SCD1	Souris	TGACCAcaggTAACCT		11,7

Table 1. Séquences de référence utilisées pour construire les motifs LXR

Position	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**
A	0	0	11	2	0	1	3,25	3,25
C	3	0	2	11	13	0	3,25	3,25
G	0	13	0	0	0	8	3,25	3,25
T	10	0	0	0	0	4	3,25	3,25
Position	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
A	3,25	3,25	0	9	13	0	0	2
C	3,25	3,25	0	0	0	12	13	6
G	3,25	3,25	0	4	0	0	0	0
T	3,25	3,25	13	0	0	1	0	5

Table 2. Matrice mLXR, les poids majoritaires sont soulignés

Les occurrences sont ainsi classées en fonction de leur score, le maximum possible étant celui du consensus des lettres ayant le plus de poids pour chaque position. Pour la matrice mLXR, cela signifie qu'on évalue le score par rapport à un maximum correspondant à la séquence consensus *TGACCGnnnnTAACCC*, qui est d'ailleurs présente dans 3 séquences de référence (FASN Souris, Rat et Humain). Toutes les autres séquences auront un score inférieur à ce maximum, en fonction du poids associé à chaque nucléotide pour chaque position. Pour mLXR (cf TABLE 1), l'amplitude des scores possibles est de -31,7 à 13,7 avec RSAT.
Si toutes les séquences possèdent virtuellement un score au regard d'une matrice, la plupart des outils filtrent spontanément les plus bas afin de ne conserver que des occurrences ayant un minimum de ressemblance avec la matrice. C'est ce que l'on qualifiera de *filtre par défaut* par la suite. Dans notre cas, avec le logiciel RSAT, le seuil par défaut pour mLXR est fixé à 6,6. Ainsi, si l'utilisateur ne fixe pas son propre seuil, les occurrences totalisant un score inférieur à 6,6 ne seront pas retenues dans les fichiers de résultats.
Pour établir le seuil par défaut, le logiciel RSAT liste toutes les valeurs théoriques possibles permises par la matrice puis calcule la Pvalue associée à chacun de ces scores. Les mots les plus proches du consensus ont les scores les plus élevés, mais aussi les plus rares : ils bénéficient donc des Pvalues les plus intéressantes. Un filtre sur la Pvalue des scores permet ainsi d'éliminer les mauvais scores. Par défaut, RSAT propose une Pvalue de 10e-4, ce qui correspond au score de 6,6 pour mLXR. Ce filtre permettant de conserver toutes les références tout en éliminant les séquences les plus fantaisistes par rapport à la matrice, nous l'avons conservé comme filtre par défaut. Exemple de séquence aux limites (i.e. ayant un score de 6,6) : *CGACCAtttcTGATCA*. Pour d'autres outils tels que JASPAR [9] et oPOSSUM [6], la valeur proposée pour le score seuil est fixée à 80-85% du score maximum permis par la matrice. Appliqué à mLXR, un tel seuil serait entre 5 et 7.

2.3 Expression régulière LXR et pattern pLXR : Le fonctionnement des expressions régulières est plus explicite : les nucléotides de chaque référence sont rassemblés pour définir les nucléotides possibles à chaque position. Voici l'expression régulière du motif LXR : *[CT]G[AC][AC]C[AGT]nnnnT[AG]A[CT]C[ACT]*. Dans

cette représentation, les lettres entre crochets indiquent toutes les possibilités différentes pour une position et n signifie [ACGT]. De cette façon, *TGACCGacgtTAACCC* et *CGCCCGacgtTAACCT* sont tous deux des occurrences reconnues par l'expression régulière, car ils vérifient une alternative possible à chaque position. Par contre, la séquence ***G****GACCGacgtTAACCC*, pourtant proche du consensus, sera refusée puisqu'elle ne vérifie aucune des alternatives possibles en position 1.
Par ailleurs, il n'y a pas de notion de score pour les expressions régulières : ainsi, les séquences *TGACCGacgtTAACCC* et *CGCACAacgtTGATCA* seront acceptées de la même façon, alors que la première est la séquence consensus et que la seconde cumule tous les choix minoritaires à chaque position.
Une des particularités de l'expression régulière réside dans son rejet systématique d'une séquence qui diffère légèrement des alternatives qu'elle autorise. Un des moyens d'élargir la recherche est d'ajouter un modèle d'erreur. Il définit un certain "degré de liberté", c'est-à-dire la possibilité de s'écarter de l'expression régulière en ne respectant pas les choix d'une ou plusieurs positions. On parle alors d'"expression régulière approchée" [3,7,10]. Nous avons ainsi défini le "pattern pLXR" (cf FIG. 1) : il représente tous les mots acceptés par l'expression régulière plus ceux qui s'en approchent à 1 ou 2 substitutions près.

(*[CT]G[AC][AC]C[AGT]nnnnT[AG]A[CT]C[ACT]*) : [0,2] substitutions

Figure 1. Pattern pLXR : expression régulière du motif LXR, à 2 substitutions près

3. Jeux de séquences analysés

a. Au voisinage de l'expression régulière : les séries ER

Afin de comparer les cibles obtenues par l'expression régulière autorisant un certain degré de liberté avec les cibles que retient la matrice poids-position, nous avons construit une série de 3 jeux disjoints de séquences : ER0, ER1 et ER2.
Le jeu de données ER0 regroupe toutes les séquences qui respectent strictement les possibilités permises par l'expression régulière pour chaque position (par exemple : *CGCACAacgtTGATCA*). Il a été généré par l'utilisation de toutes les combinaisons de choix possibles pour toutes les positions, ce qui aboutit à un ensemble de 73 728 séquences (c'est-à-dire 2x2x2x3x4^4x2x2x3).
Le jeu de données ER1 regroupe toutes les séquences qui dérivent du jeu ER0 à un nucléotide près (par exemple : ***G****GACCGacgtTAACCC*). Pour ce faire, à partir de chaque séquence du jeu ER0, toutes les possibilités de substitutions ont été générées et toutes les séquences obtenues qui n'étaient pas déjà présentes dans le jeu ER0 ont été conservées. Le jeu ER1 regroupe un total de 1 523 712 séquences différentes.
Le jeu de données ER2 regroupe de la même façon toutes les séquences qui dérivent du jeu ER1 à un nucléotide près, en rejetant les séquences déjà contenues dans les jeux ER0 ou ER1. De cette façon, les séquences obtenues dérivent de l'expression régulière à deux substitutions près (par exemple : ***A****GCAC****C****acgtTGATCA*). Le jeu ER2 regroupe un total de 15 893 632 séquences.

b. Au voisinage des références : les séries Ref

Les séquences de la série ERx ainsi dérivées de l'expression régulière ne ressemblent pas forcément aux séquences de référence. À côté de ces séquences "théoriques", nous avons donc créé une autre série rassemblant les séquences voisines des séquences de référence. Il s'agit de la série Refx, constituée de 3 jeux de données : Ref0, Ref1 et Ref2.
Le jeu de données Ref0 rassemble simplement les 13 séquences de référence (exemple : *TGAACTtgggTGACCA*). Le jeu de données Ref1 rassemble toutes les possibilités de variants d'une séquence Ref0 avec une substitution, si ce variant n'appartient pas déjà au jeu Ref0 (exemple : *TGA****G****CTtgggTGACCA*). De la même façon, le jeu Ref2 rassemble toutes les possibilités de variants d'une séquence Ref1 avec une substitution, si ce variant n'appartient pas déjà au jeu Ref0 ou Ref1, ce qui aboutit à toutes les variants de Ref0 avec 2 substitions (exemple : *TGA****G****CTtgg****a****TGACCA*).
Le jeu de données Ref0 contient 13 séquences, le jeu Ref1 en contient 648 et le jeu Ref2 en contient 17 842.

c. Modalité de test d'un jeu de données

Les jeux de données des différentes séries ont été soumis à la matrice mLXR, au moyen du logiciel RSAT, avec un bruit de fond nucléotidique neutre (équiprobabilité de présence d'un nucléotide à chaque position) et en laissant le seuil par défaut. Les scores des occurrences obtenues s'étalent donc de 6,6 à 13,7. Le choix du bruit de fond s'est posé. Nous aurions par exemple pu choisir de faire les tests dans le contexte des séquences promotrices humaines, où les probabilités de présence sont de l'ordre de 27% pour a et t, et 23% pour c et g. Après quelques tests, il semble que ce choix n'apporte qu'un léger biais à l'étude.

4. Quelle répartition des séquences de la série Refx dans la série ERx ?

Par construction, toutes les séquences de la série Refx sont incluses dans la série ERx. En particulier, les Ref0 sont incluses dans ER0. Il est néanmoins intéressant d'observer de quelle façon les Refx se répartissent au sein des jeux ERx (cf TABLE 3).

	ER0	**ER1**	**ER2**
Ref0 (13 sequences)	100% (13 sequences)	-	-
Ref1 (648 sequences)	54,2% (351 séquences)	45,8% (297 séquences)	-
Ref2 (17 842 sequences)	29,05% (5 183 séquences)	50,55% (9 018 séquences)	20,4% (3 641 séquences)

Table 3. Répartition des séquences de la série Refx au sein de la série ERx

L'analyse de la répartition des séquences de la série Refx permet de remarquer que seule la moitié (54%) des séquences Ref1 est acceptée par l'expression régulière (ER0). Par ailleurs, en autorisant une substitution dans l'expression régulière (ER0 $\cup$ ER1), on constate que la majorité des séquences proches des références est acceptée. En effet, toutes les séquences Ref0 et Ref1 et 80% des Ref2 sont acceptées par ER0 ou ER1. Ainsi, dans notre cas, *l'expression régulière avec 1 substitution apparaît suffisante pour accepter un voisinage raisonnable des séquences de référence*.

5. Combien de séquences des séries ERx et Refx passent le filtre par défaut de RSAT ?

Avec son seuil par défaut, RSAT rejette automatiquement les occurrences dont le score est trop faible et qui s'écartent donc trop de la matrice. Une de nos questions consiste à savoir combien d'éléments de chacun des jeux des différentes séries sont retenus par ce filtre standard.

	Jeu ER0	Jeu ER1	Jeu ER2
Séquences en entrée	73 728	1 523 712	15 893 632
Séquences acceptées	63 488	351 744	3 584
Pourcentage d'acceptation	86,11%	23,08%	0,02%
Pourcentage de la population retenu par le filtre	15,15%	83,9%	0,8%
	Jeu Ref0	Jeu Ref1	Jeu Ref2
Séquences en entrée	13	648	17 842
Séquences acceptées	13	585	12 103
Pourcentage d'acceptation	100%	90%	67,8%
Pourcentage de la population retenu par le filtre	0,1%	4,6%	95,2%

Table 4. Quantité de séquences acceptées par la matrice LXR pour les séries ERx et Refx

Il ressort de nos tests (cf TABLE 4) que les séquences ER0 passent presque toutes mais que 14% des séquences sont tout de même rejetées (i.e. *14% des séquences n'ayant pourtant que des 'positions autorisées' sont cependant rejetées*). Les ER1 sont minoritairement (23%) acceptées et quasiment aucune ER2 (0,02%). Par ailleurs, sur la quantité globale de séquences acceptées, les ER0 ne représentent qu'une faible partie (15%) en raison des volumes de départ différents des jeux de données ERx.
On remarque par ailleurs qu'en toute logique, les séquences proches des références (série Refx) passent facilement le filtre automatique. Cela dit, toutes les séquences très proches des références ne passent pas le filtre. Ainsi, *10% des Ref1 sont directement rejetées, c'est un résultat peu intuitif pour une variation mineure d'une mutation*. (exemple de séquence rejetée : *TGAACTtgagTGACTA*).

6. À quoi ressemblent les séquences des séries ERx et Refx acceptées par le filtre par défaut ?

Les séquences des jeux de données ER0, ER1 et ER2 ont été analysées afin de caractériser la raison pour laquelle elles passaient ou non le filtre automatique. Ce sont logiquement les poids associés à chaque position de la matrice qui ont permis de déterminer le destin de chaque séquence (cf TABLE 2).
Les séquences ER0 qui cumulent de nombreux choix faibles, par exemple : *CGCACAnnnnTGATCA*, et qui ressemblent finalement peu aux séquences de référence se retrouvent automatiquement rejetées. De façon plus précise, les séquences cumulant plus de 3 choix faibles (i.e. ayant un poids de 2/13 ou moins) n'ont pas été retenues. Inversement, les ER1 et les rares ER2 ayant passé le filtre sont ceux qui compensent la présence d'une "valeur interdite" par la présence du meilleur choix possible à de nombreuses autres positions. Par exemple, *TGACCGacgtTAACC**G*** chez les ER1 ou *TGACCGacgtTAAC**AG*** chez les ER2.
Ces résultats suggèrent qu'*a contrario* de l'expression régulière, qui accepte toutes les possibilités permises, la matrice poids-position filtre raisonnablement les mauvaises occurrences (c'est-à-dire les occurrences qui s'écartent fortement des références). Ainsi 20% des séquences ER1 et la quasi-totalité des séquences ER2 sont éliminées par le filtre par défaut alors que dans le même temps, seuls 10% des séquences Ref1 et et 30% des séquences Ref2 sont rejetées. *Les séquences Refx, proches des références, passent donc finalement mieux le filtre automatique que l'ensemble des séquences ERx, proche de l'expression régulière*.

Au vu de ces résultats, il est intéressant d'observer la répartition des séquences Refx ayant passé le filtre de la matrice au sein aux séquences ERx (cf TABLE 5).

Ref acceptées par la matrice	**ER0**	**ER1**	**ER2**
Ref0 (13/13 seq., 100%)	100% (13 seq.)	-	-
Ref1 (585/648 seq., 90%)	60% (351 séq.)	40% (234 séq.)	-
Ref2 (12 103/17 842 seq., 67%)	42,77% (5 177 séq.)	56,95% (6 892 séq.)	0,28% (34 séq.)

Table 5. Répartition au sein de la série ERx des séquences Refx acceptées par la matrice (avec filtre par défaut)

En comparant avec la répartition initiale (cf TABLE 3), on constate que la plupart des Ref2 de type ER2 ont été rejetées et que la quasi-totalité des Refx de type ER0 ont été conservées.

7. Comment se répartissent les séquences des séries ERx et Refx en termes de score ?

Bien que chaque jeu de données ER0, ER1 et ER2 ait des éléments ayant passé le filtre automatique de RSAT, la question demeure de savoir si les occurrences possèdent le même type de score en fonction de leur nature (respectant la matrice ou s'en écartant) et si elles sont davantage groupées autour de score-clef ou, à l'inverse, réparties sur tout le spectre possible des scores. Pour le vérifier, l'ensemble des scores des occurrences ERx a été porté sur un boxplot (cf FIG. 2.A).

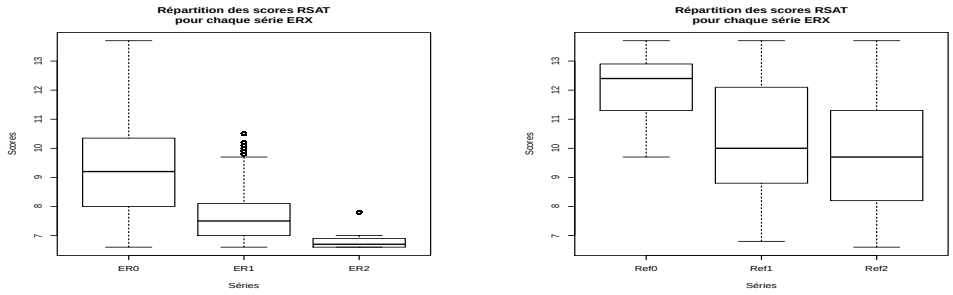

Figure 2. Boxplot des scores RSAT. FigA : pour les séquences ER0, ER1 et ER2. FigB : pour Ref0, Ref1 et Ref2

Ce diagramme nous montre que les scores du jeu de données ER0 couvrent l'ensemble des scores possibles, tandis que les scores des jeux de données ER1 et ER2 sont davantages ramassés vers les scores minima. On s'attendait bien à ce que du fait de leurs mutations, qui affectent négativement le score des séquences des jeux ER1 et ER2, les séquences ER1 et ER2 aient de mauvais scores. Par contre, la répartition régulière des scores ER0 est plus étonnante : on s'attendait à ce que les scores soient massivement favorables. Ainsi, on constate

que *les séquences respectant l'expression régulière stricte se répartissent dans des scores très variés, y compris très mauvais*.

Une question sous-jacente qui se pose est de savoir à quoi ressemblent les occurrences aux scores clé de ce box-plot. Autour de la médiane du jeu de données ER0 (score de 9,2), on trouve des occurrences qui peuvent cumuler jusqu'à 4 choix non-optimaux voire 2 choix faibles (poids de 2 ou 1) par rapport à la matrice, tel que ***C**GAA**C**A**A**acgtTGA**T**CT*. Dans le même temps, ce score correspond aussi à des occurrences ER1, composées d'une mutation et d'un choix faible (poids de 1/13), tel que *TGACC**A**gaagTAACC**G***, ou bien une mutation et deux choix non-optimaux, tel que *TGACCGacgt**T**G**A**A**C**T*. Si on remonte jusqu'à la valeur du 3ème quartile ER0 (score de 10,35), alors les séquences ER0 comptent jusqu'à 3 choix non-optimaux, tel que ***C**GA**A**CAacgtTAACCT*.
En remontant le score de validation à 10,35 (quartile3 des scores du jeu ER0), la majorité des occurrences ERx obtenues s'approche du consensus majoritaire et ces occurrences ressemblent donc fortement aux références. Inversement, en descendant à un score de validation de 8 (quartile1 des scores du jeu ER0), on obtient des occurrences plus divergentes du consensus, parmi lesquelles de nombreuses occurrences possédant un choix non-autorisé par la matrice (25% des occurrences ER1 ont un score supérieur à 8,1). La connaissance des types d'occurrences obtenues en fonction du score permet donc de choisir plus efficacement un seuil de coupure pour sélectionner les séquences par rapport à leur proximité à la matrice, ce que nous faisons en section 8.

Enfin, sans entrer dans les détails, on remarque que *les meilleures séquences à 2 degrés de liberté des références sont intégralement comprises dans ce qui est permis par l'expression régulière stricte* cf Fig1 et 2.B.

8. Comment fixer un score utilisateur pertinent ?

Le filtre par défaut élimine automatiquement les occurrences les moins pertinentes vis-à-vis de la matrice, mais il revient à l'utilisateur de déterminer où placer la limite pour valider ou non un hit en fonction de ses besoins, c'est-à-dire de décider à quel point il accepte de s'éloigner de ses références. Dans notre cas, on considère que les hits Ref2 sont potentiellement intéressants : on cherche donc à fixer un score utilisateur qui permette d'accepter la plupart des Ref1 et les meilleurs Ref2. Deux valeurs seuils paraissent alors intéressantes : le quartile1 des scores Ref1 (score de 8,8) et le quartile1 des scores Ref2 (score de 8,2).

	Ref0	Ref1	Ref2	Refx
Quantité post-filtre auto	13	585	12 103	12 701
Quantité post-filtre 8,8	13 (100%)	440 (75%)	7 507 (62%)	7 960 (62%)
Quantité post-filtre 8,2	13 (100%)	502 (85%)	9 115 (75%)	9 630 (75,8%)
	ER0	ER1	ER2	ERx
Quantité post-filtre auto	63 488	351 744	3 584	418 816
Quantité post-filtre 8,8	37 120 (58%)	45 568 (13%)	0	82 688 (19,7%)
Quantité post-filtre 8,2	45 312 (71%)	80 384 (23%)	0	125 969 (30%)

Table 6. Quantité de séquences des séries Refx et ERx en fonction de différents filtres utilisateurs

L'étude de la répartition des Refx et des ERx donne donc des indications précieuses pour fixer un seuil utilisateur pertinent. Par exemple ici, il semble adéquat de choisir un filtre utilisateur de 8,8 car il constitue un compromis qui permet de récupérer la majorité des séquences Ref1 (75%) et Ref2 (62%) tout en minimisant la quantité de séquences ER1 (13%) qui sont moins proches de nos références, tandis qu'un filtre utilisateur de 8,2 récupérerait davantage de Ref1 (85%) et Ref2 (75%) mais aussi près du double de séquences ER1 (23%). On note au passage qu'avec les deux scores proposés (8,8 et 8,2), aucune séquence du jeu de données ER2 n'est retenue, alors qu'on garde une grande partie des séquences Ref2.

9. Conclusion

Au travers d'un cas d'étude, la recherche du TFBS de LXR, nous avons comparé les occurrences retenues par une matrice de score avec celles retenues par une expression régulière. Nous cherchions à savoir de quelle façon ces deux méthodes s'acquittent de leur mission : trouver les occurrences proches des séquences de référence obtenues expérimentalement. Nous voulions aussi trouver un critère pour fixer un score seuil d'accep-

tation de la matrice. Pour cela, nous avons construit deux séries de données : ERx, qui regroupe les séquences autour de l'expression régulière jusqu'à deux substitutions près, et Refx, qui regroupe les séquences dérivant des références, jusqu'à deux substitutions près, c'est-à-dire l'essentiel des séquences que nous voudrions cibler.

Voici les principaux résultats obtenus sur l'étude LXR. Concernant les expressions régulières : lorsque le consensus permet de nombreuses alternatives (dans le cas du LXR, seules 4 positions sur 16 sont à choix unique), le motif obtenu peut s'avèrer très flou (74 000 occurrences pour l'expression régulière LXR). L'explosion combinatoire devient même phénoménale si on autorise 1 événement de substitution (+ 1,5 million d'occurrences) ou 2 (+ 16 millions !). Par ailleurs, dans le cas LXR, la majorité (80%) des séquences proches des références (i.e. les Refx, jusqu'à 2 substitutions d'une référence) est couverte par l'expression régulière avec au plus 1 substitution (i.e par ER0 et ER1). Il ne semble donc vraiment pas pertinent dans ce type de cas d'étendre la recherche jusqu'à 2 substitutions. En effet, cela génère énormément de faux positifs et quasiment aucun positif. Et même au-delà, on peut dire que si trop de positions du motif permettent des alternatives, alors les expressions régulières, même strictes, ne sont pas adaptées.

Concernant les matrices de scores : il apparaît que les séquences Refx, proches des références biologiques, passent plus facilement le filtre automatique de la matrice que les séquences ERx : ainsi, si seules 20% des séquences couvertes par l'expression régulière avec 1 substitution sont conservées, 90% et 66% des séquences dérivant des références avec respectivement 1 et 2 substitutions sont malgré tout conservées. En outre, les séquences Refx disposent globalement de scores plus élevés que les séquences ERx. On a ainsi pu positionner le score utilisateur à une valeur permettant de récupérer la majorité des séquences proches des références tout en minimisant la quantité de faux positifs. Dans notre cas LXR, la matrice de score se révèle donc mieux adaptée que les expressions régulières à la détection de nouveaux sites candidats. Par ailleurs, on constate que l'étude de la répartition des Refx et des ERx donne des indications déterminantes pour fixer un seuil utilisateur pertinent.

Cette étude, sur un cas particulier, vise à apporter une meilleure perception du champ d'action des méthodes de *pattern matching* par score et par expression régulière, dans le cas de motifs simples tels que les TFBS. Cependant, elle n'a que valeur d'exemple. En particulier, l'étude pourrait être complétée en étudiant de plus près, de façon expérimentale ou théorique, l'influence sur les résultats de la taille des séquences de références, de leur homogénéité et notamment de l'importance du nombre de positions à choix unique. Le but en serait de fournir des règles générales, utilisables en pratique, pour évaluer la pertinence d'utiliser ou non des expressions régulières - et avec quel degré de liberté - ainsi que pour aider à fixer le seuil d'acceptation de la matrice.

Références

[1] P. Rice, I. Longden and A. Bleasby. EMBOSS : the European Molecular Biology Open Software Suite. *Trends in genetics*, 16(6) :276–277, Jun 2000.

[2] G. Kucherov and M. Rusinowitch. Matching a set of strings with variable length don't cares. *Springer Berlin Heidelberg*, Combinatorial Pattern Matching :230-247, 1995.

[3] G. Navarro. NR-grep : a fast and flexible pattern-matching tool. *Software : Practice and Experience*, 31(13), :1265-1312, 2001.

[4] J. Schug and G. Overton. TESS : Transcription Element Search Software. *WWW Technical Report CBIL-TR-1997-1001-v0.0, of the Computational Biology and Informatics Laboratory*, 1997.

[5] J.-V. Turatsinze, M. Thomas-Chollier, M. Defrance, and J. van Helden. Using RSAT to scan genome sequences for transcription factor binding sites and cis-regulatory modules. *Nature Protocols*, 3(10) :1578–1588, 2008.

[6] A. T. Kwon, D. J. Arenillas, R. Worsley Hunt and W. W. Wasserman. oPOSSUM-3 : advanced analysis of regulatory motif over-representation across genes or ChIP-Seq datasets. *G3 (Bethesda, Md.)*, 2(9) :987-1002, Sep.2012.

[7] P. Muzátko. Approximate Regular Expression Matching. Stringology, :37-41, 1996.

[8] O. Demeure, F. Lecerf, C. Duby, C. Desert, S. Ducheix, H. Guillou, and S. Lagarrigue. Regulation of LPCAT3 by LXR. *Gene*, 470(1-2) :7–11, Jan. 2011.

[9] A. Mathelier and W. W. Wasserman. The Next Generation of Transcription Factor Binding Site Prediction. *PLoS Computational Biology*, 9(9), Sep.1997.

[10] E. W. Myers and W. Miller. Approximate matching of regular expressions. *Bulletin of mathematical biology*, 51(1) :5-37, 1989.